Reed-Solomon CODEC Megafunction

Solution Brief 1 **October 1996, ver. 1**

Target Application:
Communications
Digital Signal Processing

Family:
FLEX® 10K

Vendor:

Object Oriented Hardware

10-16 Tiller Road
Docklands, London E14 8PX
England, UK
Tel. 44 (0) 171 538 5858
Fax 44 (0) 7000 664329
Fax 44 (0) 171 538 2323
WWW http://www.ooh.com

Features

- Foundry Independent Standard Product (FISP)
- Fully parameterized
- Corrects up to 2*n* erasures or *n* errors per block
- Continuous or burst-mode operation
- Programmable generator and primitive polynomials
- Complies with Intelsat IESS-308, Revision 6B or RTCA DO-217 Appendix F, Revision D
- Independent encoding and decoding
- Statistics and error rate gathering options
- Fully synthesizable VHDL-RTL code
- Interface ports
  - Support for serial and parallel data formats
  - FISPbus generic microprocessor interface
- Applications
  - Satellite communications
  - Digital video
  - Magnetic and optical tape and disk drives
  - High-performance modems
  - Local and wide area networks

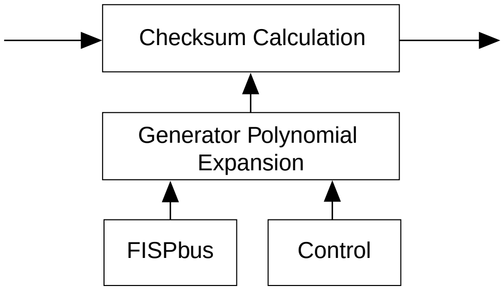
General Description

The Reed-Solomon encoder/decoder (CODEC) megafunction provides a complete solution for encoding and decoding data. The CODEC megafunction also provides statistical information about the number of correctable and uncorrectable errors that occur over the decoder channel. This megafunction is implemented in VHDL and is optimized for the Altera® FLEX 10K device architecture.

Reed-Solomon Encoder

The encoder receives raw data and adds check symbols. The codeword length and the number of check symbols are assigned using the FISPbus interface or can be specified via direct inputs to the data block. The output produces the original data with the check symbols appended. The delay between input and output depends directly on the number of check symbols and the clocking mode. See Figure 1.

Figure 1. Reed-Solomon Encoder

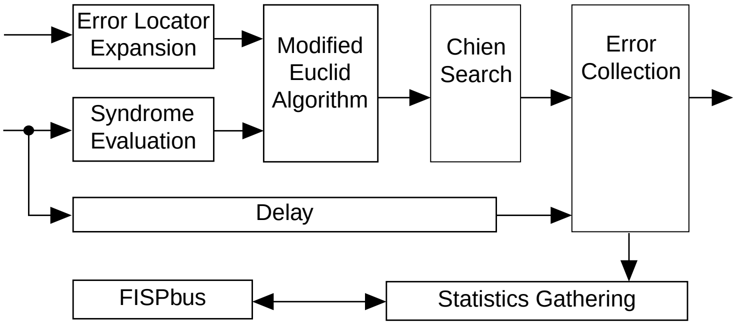


Reed-Solomon Decoder

The Reed-Solomon decoder can correct up to *n* symbol errors or 2*n* erasures in each data block. The data received can be continuous, and the delay between input and corrected output is independent of the number of errors or erasures. The time required to perform the correction depends on the codeword length and the number of check bits. The decoder accepts data in either serial or parallel data formats; the start of the block and any erasure symbols are flagged. The decoder sequence,shown in Figure 2, is summarized below:

1. Syndrome and error locator polynomials are generated.
2. Euclid algorithm is performed.
3. Modified Chien search is performed.
4. Statistical information is generated.

Figure 2. Reed-Solomon Decoder



Reed-Solomon CODEC Parameters

Table 1 summarizes the parameters that can be defined by the user. Contact Object Oriented Hardware (OOH) for information on choosing the best parameter settings to meet the requirements of specific applications.

Table 1. Reed-Solomon CODEC Parameters

Parameter	Description
Number of symbols in the longest codeblock	Expressed as two figures, normally written (*n*, *k*), where *n* is the number of symbols in the codeblock after check bits are added, and *k* is the original data length. The largest value of *n* sets the required size of the Galois Field.
Symbol width	Number of required bits to represent the maximum codeblock length in binary.
Required check bits and differing values of *n* and *k*	Each corrected error requires two check bits. For correction codes, this value is the maximum value of *n* and *k*. A small number of check bits reduces the encoder size.
Primitive polynomial	Normally expressed in the form: $P(x) = x^m + x^a + x^b + x^c + ... + 1$
Generator polynomial	The generator polynomial is shown below (where R is the number of check bits): $$G(x) = \prod_{j=B}^{(B-1)+R} (x - \alpha^j)$$
Number of erasures	Errors at known positions. Some applications force errors to these known positions to transfer additional data. If erasures are not used, the decoder can be simplified. An erasure can be corrected for each check bit, but often only a few erasures are required, which saves significant space in the decoder. Erasures at fixed positions (i.e., end of codeblock) enable additional logic savings.
System clock frequency	High clock frequencies reduce the latency (and hence the size of the delay line) in the decoder.
Statistic gathering functions	Typically counters for errors corrected, uncorrectable codeblocks, and codeblocks received.
Serial or parallel data interfaces	Internal data paths are parallel. If serial interfaces are needed, shift registers can be added.